



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

# TP1

---

## Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Sebastián Fernandez Ledesma		
Fernando Gasperi Jabalera	56/09	fgasperijabalera@gmail.com
Maximiliano Wortman		
Santiago Camacho	110/09	santicamacho90@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
**Universidad de Buenos Aires**

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

<b>1. Problema 1: Puentes sobre lava caliente</b>	<b>3</b>
1.1. Presentación del problema . . . . .	3
1.2. Resolución . . . . .	3
1.2.1. Algoritmo . . . . .	3
1.2.2. Pseudocódigo . . . . .	3
1.3. Demostración . . . . .	3
1.4. Análisis de complejidad . . . . .	3
1.5. Test de complejidad . . . . .	3
1.6. Testing . . . . .	3
<b>2. Problema 2: Horizontes lejanos</b>	<b>4</b>
2.1. Presentación del problema . . . . .	4
2.2. Resolución . . . . .	4
2.2.1. Algoritmo . . . . .	4
2.2.2. Pseudocódigo . . . . .	4
2.3. Demostración . . . . .	4
2.4. Análisis de complejidad . . . . .	4
2.5. Test de complejidad . . . . .	4
2.6. Testing . . . . .	4
<b>3. Problema 3: Biohazard</b>	<b>5</b>
3.1. Presentación del problema . . . . .	5
3.2. Resolución . . . . .	5
3.2.1. Algoritmo . . . . .	5
3.2.2. Pseudocódigo . . . . .	5
3.3. Demostración . . . . .	5
3.4. Análisis de complejidad . . . . .	5
3.5. Test de complejidad . . . . .	5
3.6. Testing . . . . .	5

## 1. Problema 1: Puentes sobre lava caliente

### 1.1. Presentación del problema

### 1.2. Resolución

#### 1.2.1. Algoritmo

#### 1.2.2. Pseudocódigo

### 1.3. Demostración

Vamos a definir un salto como un entero natural mayor que 0 y menor que la cantidad de tablones que el participante actual puede saltar de a una sola vez. Nuestra implementación recorre el puente dando saltos, garantizando que en cada salto, la distancia recorrida es máxima. Es decir, no existe otro salto tal que la distancia desde donde estamos parados es mayor a la del salto actual y el tablón en el que caes no está roto.

*Distancia* es un  $\text{Nat} > 0$ .

*Salto* es  $\text{Nat}$  tal que  $\forall s : \text{Salto}, s > 0 \wedge s \leq \text{Distancia}$

Vamos a tratar de probar que dado una secuencia de saltos, si para cada salto  $s$ ,  $s$  es un "salto máximo" si la sumatoria de saltos es mayor a la cantidad de tablones del puente, entonces nuestra secuencia es solución del problema.

Dada un  $\text{Sec}(\text{Salto})$   $se$ .

$$(\forall i : \text{Nat}, i < se.\text{long})(esMax(se_i) \wedge \sum_{j=0}^{se.\text{long}-1} se_j = \text{puente.long}) \implies \\ \nexists (se' : \text{Sec}(\text{Salto})) / se.\text{long} < se'.\text{long} \wedge \sum_{j=0}^{se'.\text{long}-1} se'_j \geq \text{puente.long} //$$

Llamemos a  $sMax$  a la secuencia de saltos máximos obtenida. Supongamos que existe secuencia  $s$  de saltos tal que la cantidad de elementos de  $s$  es menor a  $sMax$  y la sumatoria de saltos es igual o mayor a la cantidad de tablones. Bueno en particular, existe al menos un salto  $s_i$ , tal que  $s_i$  es mayor a  $sMax_i$ , ya que si todos los  $s_i$  son menores a su correspondiente  $sMax_i$ , entonces la sumatoria de  $sMax$  es mayor que la sumatoria de  $s$ . (comprobar esto ad-hoc, probablemente sale por inducción). Bueno, supongamos que agarré el primero de todos los  $s_i$ , que es más grande que su correspondiente  $sMax_i$ . Hasta ese momento las dos subsecuencias (desde el principio hasta el elemento  $i$ ) pesan lo mismo, entonces  $s_i$  está parado en el mismo lugar y hace un salto más grande que el salto máximo ( $sMax_i$ ), lo cual es absurdo. Por lo tanto queda comprobado que ese  $s_i$ , no puede existir y la solución es máxima.

### 1.4. Análisis de complejidad

### 1.5. Test de complejidad

### 1.6. Testing

## 2. Problema 2: Horizontes lejanos

### 2.1. Presentación del problema

### 2.2. Resolución

#### 2.2.1. Algoritmo

#### 2.2.2. Pseudocódigo

---

```
input: edificios
while quedan edificios do
  if empieza edificio then
    registro el edificio como abierto
    if altura del edificio es mayor a la del contorno then
      agrego la altura del edificio al contorno
    end if
  else
    saco al edificio de los abiertos
    if este edificio le daba la altura al contorno then
      agrego la altura del edificio abierto que le siga en altura al contorno
    end if
  end if
end while
return contorno
```

---

### 2.3. Demostración

### 2.4. Análisis de complejidad

### 2.5. Test de complejidad

### 2.6. Testing

### **3. Problema 3: Biohazard**

#### **3.1. Presentación del problema**

#### **3.2. Resolución**

##### **3.2.1. Algoritmo**

##### **3.2.2. Pseudocódigo**

#### **3.3. Demostración**

#### **3.4. Análisis de complejidad**

#### **3.5. Test de complejidad**

#### **3.6. Testing**